

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Scott J. Broussard, Ying Liu, Eduardo N. Spring  
Assignee: International Business Machines Corporation  
Title: Method and System for Grid-Enabled Virtual Machines With Distributed Management of Applications  
Serial No.: 10/762,000 Filing Date: January 21, 2004  
Examiner: Jennifer N. To Group Art Unit: 2195  
Docket No.: AUS920030872US1 Customer No.: 65362

July 16, 2009

*Filed Electronically*

**APPEAL BRIEF UNDER 37 CFR § 41.37**

Dear Sir:

Applicant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on May 21, 2009, and the Notice of Panel Decision mailed June 3, 2009. The fee for this Appeal Brief is being paid electronically via the USPTO EFS. The Board is authorized to deduct any other amounts required for this appeal brief and to credit any amounts overpaid to Deposit Account No. 090447.

**I. REAL PARTY IN INTEREST - 37 CFR § 41.37(c)(1)(i)**

The real party in interest is the assignee, International Business Machines Corporation, as named in the caption above and as evidenced by the assignment set forth at Reel 014636, Frame 0146.

**II. RELATED APPEALS AND INTERFERENCES - 37 CFR § 41.37(c)(1)(ii)**

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

### **III. STATUS OF CLAIMS - 37 CFR § 41.37(c)(1)(iii)**

Claims 1, 5-9, 13-17, and 21-24 are pending in the application. Claims 2-4, 10-12, and 18-20 are canceled. Claims 1, 5-9, 13-17, and 21-24 stand rejected. The rejection of claims 1, 5-9, 13-17, and 21-24 is appealed. Appendix “A” contains the full set of pending claims.

### **IV. STATUS OF AMENDMENTS - 37 CFR § 41.37(c)(1)(iv)**

No Amendments have been filed subsequent to final rejection.

### **V. SUMMARY OF CLAIMED SUBJECT MATTER - 37 CFR § 41.37(c)(1)(v)**

The present invention, as set forth by independent claim 1 relates to a method for operating a virtual machine within a data processing system (see e.g., ¶ 0044, lines 1-4). The method includes the computer-implemented steps of running a plurality of virtual machines on one or more devices within the data processing system (see e.g., ¶ 0044, lines 4-8), wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machines in a virtual machine cluster (see e.g., ¶ 0045, lines 1-10), associating the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster (see e.g., ¶ 0047, lines 1-10), and sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run (see e.g., ¶ 0049, lines 1-10).

The present invention, as set forth by independent claim 9 relates a computer program product on a computer readable medium for use within a data processing system for operating a virtual machine (see e.g., ¶ 0142). The computer program product includes means for running one of a plurality of virtual machines on one or more devices within the data processing system (see e.g., ¶ 0044, lines 4-8), wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machines in a virtual machine cluster (see e.g., ¶ 0045, lines 1-10), means for associating a virtual machine with the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster ( see e.g., ¶ 0047, lines 1-10), and means for sharing information about the plurality of virtual machines within the virtual

machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run (see e.g., ¶ 0049, lines 1-10).

The present invention, as set forth by independent claim 17 relates to an apparatus within a data processing system for operating a virtual machine (see e.g., ¶ 0044, lines 1-4). The apparatus includes means for running one of a plurality of virtual machines on one or more devices within the data processing system (see e.g., ¶ 0044, lines 4-8), wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machines in a virtual machine cluster (see e.g., ¶ 0045, lines 1-10), means for associating a virtual machine with the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster ( see e.g., ¶ 0047, lines 1-10), and means for sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run (see e.g., ¶ 0049, lines 1-10).

## **VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 9 and 13-16 are rejected under 35 U.S.C. § 101 as directed to non-statutory matter. Claims 1, 5-9, 13-17, and 21-24 are rejected under 35 U.S.C. § 103(a) over Rietschote, U.S. Patent No. 7,203,944 (Rietschote) in view of Zhu et al., “Jessica2: A Distributed Java Virtual Machine with Transparent Thread Migration Support,” IEEE, 2002, pages 381-388 (Zhu).

## **VII. ARGUMENTS**

### **Claims 9 and 13 – 17 are allowable over 35 U.S.C. § 101.**

In the Response filed on October 29, 2008, claims 9 and 17 were amended to affirmatively set forth that the computer program product (claim 9) and the apparatus (claim 17) are executed by a data processing system. Based upon this amendment the computer program product and apparatus are more limited than the computer readable medium disclosed within the specification. Accordingly, claims 9 and 13 – 17 are statutory.

**Claims 1, 5 – 9, 13 – 17 and 21 – 24 are allowable over Rietschote and Zhu.**

Rietschote generally disclose a cluster of computer systems where the computer systems may include virtual machines. (See e.g., Rietschote, Col. 2, lines 46 – 65.). Rietschote discloses that the virtual machine resource may be failed over in response to detection of a failure. However, there is no disclosure in Rietschote of sharing information about a plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run or of determining that a CPU load utilization on a first virtual machine exceeds a threshold value and moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value. This deficiency of Rietschote is not cured by Zhu.

Zhu discloses a distributed Java Virtual Machine which includes a thread migration mechanism to enable dynamic load balancing by migrating Java threads between cluster nodes at runtime without programmers' involvement. There is no disclosure in Zhu of determining that a CPU load utilization on a first virtual machine *exceeds a threshold value and moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value.*

Accordingly, Rietschote or Zhu, taken alone or in combination, do not disclose or suggest determining that a CPU load utilization on a first virtual machine exceeds a threshold value and moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value, all as required by claim 1 and as substantially required by claims 9 and 17. Accordingly, claims 1, 9 and 17 are allowable over Rietschote and Zhu. Claims 5 – 8 depend from claim 1 and are allowable for at least this reason. Claims 13 – 16 depend from claim 9 and are allowable for at least this reason. Claims 21 – 24 depend from claim 17 and are allowable for at least this reason.

**VIII. CLAIMS APPENDIX - 37 CFR § 41.37(c)(1)(viii)**

A copy of the pending claims involved in the appeal is attached as Appendix “A.”

**IX. EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)**

None.

**X. RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)**

There are no related proceedings.

**XI. CONCLUSION**

In view of the above arguments, it is respectfully urged that the rejection of the claims should not be sustained.

CERTIFICATE OF TRANSMISSION

I hereby certify that on July 16, 2009 this correspondence is being transmitted via the U.S. Patent & Trademark Office's electronic filing system.

*/Stephen A. Terrile/*

Respectfully submitted,

*/Stephen A. Terrile/*

Stephen A. Terrile  
Attorney for Applicant(s)  
Reg. No. 32,946

## APPENDIX A - PENDING CLAIMS

1. (Previously Presented) A method for operating a virtual machine within a data processing system, the method comprising the computer-implemented steps of:

- running a plurality of virtual machines on one or more devices within the data processing system, wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machines in a the virtual machine cluster;
- associating the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster; and
- sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run, the sharing information further comprising sharing load values representing computer resource utilization among the virtual machines in the virtual machine cluster;
- performing a load-balancing operation across the virtual machine cluster;
- determining that a CPU load utilization on a first virtual machine exceeds a threshold value; and
- moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value.

2-4. (Canceled)

5. (Previously Presented) The method of claim ~~3~~ 1 further comprising:

- determining that a memory load utilization on a first virtual machine exceeds a threshold value; and
- moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation.

6. (Original) The method of claim 1 further comprising:  
moving a thread from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.
7. (Original) The method of claim 1 further comprising:  
moving a set of one or more objects from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.
8. (Original) The method of claim 1 further comprising:  
running a multi-threaded application within the virtual machine cluster; and  
dispatching threads of the multi-threaded application on different virtual machines such that execution of the multi-threaded application spans multiple virtual machines.
9. (Previously Presented) A computer program product on a computer readable medium executed by a data processing system for operating a virtual machine, the computer program product comprising:  
means for running one of a plurality of virtual machines on one or more devices within the data processing system, wherein each virtual machine in the plurality of virtual machines incorporates functionality for interoperating with other virtual machines in a the virtual machine cluster;  
means for associating a virtual machine with the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster; and  
means for sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run, the means for sharing comprising means for sharing load values representing computer resource utilization among the virtual machines in the virtual machine cluster;  
means for performing a load-balancing operation across the virtual machine cluster.  
means for determining that a CPU load utilization on a first virtual machine exceeds a threshold value; and

means for moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value.

10–12. (Canceled)

13. (Previously Presented) The computer program product of claim 9 further comprising:

means for determining that a memory load utilization on a first virtual machine exceeds a threshold value; and

means for moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation.

14. (Original) The computer program product of claim 9 further comprising:

means for moving a thread from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.

15. (Original) The computer program product of claim 9 further comprising:

means for moving a set of one or more objects from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.

16. (Original) The computer program product of claim 9 further comprising:

means for running a multi-threaded application within the virtual machine cluster; and

means for dispatching threads of the multi-threaded application on different virtual machines such that execution of the multi-threaded application spans multiple virtual machines.

17. (Previously Presented) An apparatus within a data processing system for operating a virtual machine, the apparatus being executed by a data processing system, the apparatus comprising:

means for running one of a plurality of virtual machines on one or more devices within the data processing system, wherein each virtual machine in the plurality of



virtual machines incorporates functionality for interoperating with other virtual machines in a the virtual machine cluster;

means for associating a virtual machine with the plurality of virtual machines in a virtual machine cluster, wherein each virtual machine in the virtual machine cluster acts as a node within the virtual machine cluster; and

means for sharing information about the plurality of virtual machines within the virtual machine cluster such that a virtual machine may be added to the virtual machine cluster or such that a virtual machine may be removed from the virtual machine cluster as the plurality of virtual machines continues to run, the means for sharing information comprising means for sharing load values representing computer resource utilization among the virtual machines in the virtual machine cluster;

means for performing a load-balancing operation across the virtual machine cluster

means for determining that a CPU load utilization on a first virtual machine exceeds a threshold value; and

means for moving a thread from the first virtual machine to a second virtual machine during a load-balancing operation in response to the first virtual machine exceeding the threshold value.

18-20. (Canceled)

21. (Previously Presented) The apparatus of claim 17 further comprising:

means for determining that a memory load utilization on a first virtual machine exceeds a threshold value; and

means for moving a set of one or more objects from the first virtual machine to a second virtual machine during a load-balancing operation.

22. (Original) The apparatus of claim 17 further comprising:

means for moving a thread from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.

23. (Original) The apparatus of claim 17 further comprising:  
means for moving a set of one or more objects from a first virtual machine in the virtual machine cluster to a second virtual machine in the virtual machine cluster.

24. (Original) The apparatus of claim 17 further comprising:  
means for running a multi-threaded application within the virtual machine cluster; and  
means for dispatching threads of the multi-threaded application on different virtual machines such that execution of the multi-threaded application spans multiple virtual machines.

**EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)**

None

**RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)**

There are no related proceedings.